

K 1

UUU	UUU	EEEEEEEEEeeeeee	TTTTTTTTTTTTTTTT	PPPPPPPPPPPPPPP
UUU	UUU	EEEEEEEEEeeeeee	TTTTTTTTTTTTTTTT	PPPPPPPPPPPPPPP
UUU	UUU	EEEEEEEEEeeeeee	TTTTTTTTTTTTTTTT	PPPPPPPPPPPPPPP
UUU	UUU	EEE	TTT	PPP
UUU	UUU	EEE	TTT	PPP
UUU	UUU	EEE	TTT	PPP
UUU	UUU	EEE	TTT	PPP
UUU	UUU	EEE	TTT	PPP
UUU	UUU	EEE	TTT	PPP
UUU	UUU	EEE	TTT	PPP
UUU	UUU	EEE	TTT	PPP
UUU	UUU	EEE	TTT	PPP
UUU	UUU	EEE	TTT	PPP
UUU	UUU	EEE	TTT	PPP
UUU	UUU	EEEEEEEEEeeeeee	TTT	PPPPPPPPPPPPPPP
UUU	UUU	EEEEEEEEEeeeeee	TTT	PPPPPPPPPPPPPPP
UUU	UUU	EEEEEEEEEeeeeee	TTT	PPPPPPPPPPPPPPP
UUU	UUU	EEE	TTT	PPP
UUU	UUU	EEE	TTT	PPP
UUU	UUU	EEE	TTT	PPP
UUU	UUU	EEE	TTT	PPP
UUU	UUU	EEE	TTT	PPP
UUU	UUU	EEE	TTT	PPP
UUU	UUU	EEE	TTT	PPP
UUU	UUU	EEE	TTT	PPP
UUU	UUU	EEEEEEEEEeeeeee	TTT	PPP
UUU	UUU	EEEEEEEEEeeeeee	TTT	PPP
UUU	UUU	EEEEEEEEEeeeeee	TTT	PPP

RRRRRRRR	MM	MM	SSSSSSSS	TTTTTTTT	EEEEEEEEE	SSSSSSSS	TTTTTTTT	44	44
RRRRRRRR	MM	MM	SSSSSSSS	TTTTTTTT	EEEEEEEEE	SSSSSSSS	TTTTTTTT	44	44
RR RR	RR	MMMM	MMMM	SS	TT	EE	SS	TT	44
RR RR	RR	MMMM	MMMM	SS	TT	EE	SS	TT	44
RR RR	RR	MM MM	MM	SS	TT	EE	SS	TT	44
RR RR	RR	MM MM	MM	SS	TT	EE	SS	TT	44
RRRRRRRR	MM	MM	SSSSSS	TT	EEEEEEEEE	SSSSSS	TT	4444444444	
RRRRRRRR	MM	MM	SSSSSS	TT	EEEEEEEEE	SSSSSS	TT	4444444444	
RR RR	RR	MM	MM	SS	TT	EE	SS	TT	44
RR RR	RR	MM	MM	SS	TT	EE	SS	TT	44
RR RR	RR	MM	MM	SS	TT	EE	SS	TT	44
RR RR	RR	MM	MM	SS	TT	EEEEEEEEE	SSSSSSSS	TT	44
RR RR	RR	MM	MM	SSSSSSSS	TT	EEEEEEEEE	SSSSSSSS	TT	44

LL	IIIIII	SSSSSSSS
LL	IIIIII	SSSSSSSS
LL	II	SS
LL	II	SS
LL	II	SS
LL	II	SSSSSS
LL	II	SSSSSS
LL	II	SS
LL	II	SS
LL	II	SS
LLLLLLLL	IIIIII	SSSSSSSS
LLLLLLLL	IIIIII	SSSSSSSS

```
0000    1      .IDENT  'V04-000'  
0000    78     $BEGIN  RMSTEST4,009,--RMSTEST,<XAB RMS TEST PROGRAM>,<GBL, LONG>  
0000    79     ;  
0000    80     :  
0000    81     ;  
0000    82     .ENABL  DBG  
0000    83     ;  
0000    84     ;  
0000    85     ; macros:  
0000    86     ;  
0000    87     ;  
0000    88     ;  
0000    89     ;  
0000    90     ;  
0000    91     .MACRO  TYPE STRING, ?L  
0000    92     STORE  <STRING>  
0000    93     BLBC   VERBOSITY,L  
0000    94     MOVL   #$.TMPX,CMDORAB+RAB$L RBF  
0000    95     MOVW   #$.TMPX1,CMDORAB+RAB$W RSZ  
0000    96     $PUT   RAB=CMDORAB,ERR=REPORT_ERROR  
0000    97     BSBW   ERR  
0000    98     L:    .ENDM  TYPE  
0000    99     ;  
0000   100    ;  
0000   101    :  
0000   102    ;  
0000   103    .MACRO  STORE STRING,PRE  
0000   104    .SAVE  
0000   105    .PSECT  --$RMSNAM  
0000   106    $.TMPX=.  
0000   107    PRE  
0000   108    .ASCII  %STRING%  
0000   109    $.TMPX1=-$.TMPX  
0000   110    .RESTORE  
0000   111    .ENDM  STORE ; store any carriage control info
```

```
0000 113
0000 114 ;
0000 115 ;
0000 116 .MACRO BEGIN TSTNAM
0000 117 STORE <TSTNAM>
0000 118 MOVL #SS.TMPX,BEG_DESCR+4 ; addr
0000 119 MOVL #SS.TMPX1,BEG_DESCR ; len
0000 120 BSBW BEGPUT
0000 121 .ENDM BEGIN
0000 122 .MACRO FINISH TSTNAM
0000 123 STORE <TSTNAM>
0000 124 MOVL #SS.TMPX,FIN_DESCR+4 ; addr
0000 125 MOVL #SS.TMPX1,FIN_DESCR ; len
0000 126 BSBW FINPUT
0000 127 .ENDM FINISH
0000 128 .MACRO FLDNAM
0000 129 STORE <FLDNAM>
0000 130 MOVL #SS.TMPX,FLD_DESCR+4 ; addr
0000 131 MOVL #SS.TMPX1,FLD_DESCR ; len
0000 132 BSBW FLDPUT
0000 133 .ENDM FIELD
```

```

00000000 135 .PSECT RMSTEST.GBL, LONG
0000 136 .ALIGN C0NG
0000 137 T4START:::
0000 138 T4FAB:: $FAB FNM=<TST$DISK:T4FILE.DAT;1>,-
0000 139 ORG=SEQ,-
0000 140 RFM=VFC,-
0000 141 RAT=CR,-
0000 142 FSZ=4,-
0000 143 MRS=100,-
0000 144 NAM=NAMBLK,-
0000 145 DEQ=12
0050 146 FLUSH_FAB:: $FAB FAC=<PUT,GET>,-
0050 147 FNM=<TST$DISK:T4FILE.DAT;1>,-
0050 148 NAM=NAMBLK,-
0050 149 SHR=<PUT,GET,UPI>,-
0050 150 XAB=FHCXAB
00A0 151
00A0 152
00A0 153 :
00A0 154 : attention: in order to assemble this module, t4rab and FLUSH_RAB
00A0 155 : have been put into another module, RMSTESTR
00A0 156 :
00A0 157 :
00A0 158 FHCXAB:: $XABFHC NXT=ALQXAB
00A0 159 ALQXAB:: $XABALL NXT=PROXAB,-
00CC 160 PROXAB:: DEQ=15
00CC 161 $XABPRO
00CC 162 DATXAB:: $XABDAT
0144 163 RDTXAB:: $XABRDT
0144 164 TRMXAB:: $XABTRM
0170 165 $RMSDEFEND
0170 166 SAVEPRO: EXTRA=XAB$L_SBN-4 ; 4 bytes of extra (spare) char.
01A8 167 .WORD 0 ; word to save pro in
00000024 01A8 168
01AA 01A8 169 :THESE ARE THE DATA STRUCTURES FOR DATE AND TIME XAB CHECKS
01AA 01A8 170 :
01AA 01A8 171 :
01AA 01A8 172 :
01AA 01A8 173 :
01AA 01A8 174 :
01AA 01A8 175 :
01AA 01A8 176 :
01AA 01A8 177 :
01AA 01A8 178 :
01AA 01A8 179 :
01AA 01A8 180 CDT: .ASCII / 3-MAR-1963 03:03:03.03/
01B6 00000017 01C1 181 CDTL=-CDT
01C1 182 RDT: .ASCII / 4-APR-1944 04:04:04.04/
01CD 00000017 01D8 183 RDTL=-RDT
01D8 184 EDT: .ASCII / 8-AUG-1988 08:08:08.08/
01E4 00000017 01EF 185 EDTL=-EDT
01EF 186 RDT2: .ASCII /12-DEC-1948 12:12:12.12/
01FB 00000017 0206 187 RDTL2=-RDT2

```

RMSTEST4
009

XAB RMS TEST PROGRAM

F 9

16-SEP-1984 01:47:44 VAX/VMS Macro V04-00
5-SEP-1984 04:21:52 [UETP.SRC]RMSTEST4.MAR;1

Page 4
(4)

000001AA'00000017	0206	188	CDTDEC: .LONG	CDTL,CDT	\$S
000001C1'00000017	020E	189	RDTDEC: .LONG	RDTL,RDT	\$S
000001D8'00000017	0216	190	EDTDEC: .LONG	EDTL,EDT	\$S
	021E	191	RDT2DEC:		\$S
000001EF'00000017	021E	192	.LONG	RDTL2,RDT2	\$S
	0226	193			\$S
0000	0226	194	CURRVN: .WORD	0	\$S
0000	0228	195	LEN: .WORD	0	\$S
00000243	022A	196	CMPDAT: .BLKB	25	\$S
	0243	197	CMPDATDEC:		\$S
0000022A'00000019	0243	198	.LONG	25,CMPDAT	\$S
00000000	024B	199	CURRDT: .LONG	0	\$S
00000000	024F	200	UIC: .LONG	0	\$S
0000000B	0253	201	DATLEN=11		..
00000014	0253	202	TIMLEN=20		..
	0253	203			..
	0253	204			..

; length of returned string
; has room for longest possible date
; address of current rdt string
; room to save current uic
; length of date
; length of ascii date and time

```

      0253 206 RMTTEST_4A::: RD
OFFC 0253 207 .WORD RD
      0255 208 BEGIN RD
      026A 209 <XAB TESTS> RD
      026A 210 ; RD
      026A 211 : create a file -- sys$disk:t4file.dat:1 -- with controlled attributes RD
      026A 212 : and write 1 record to it, to further control the attributes RD
      026A 213 ; RD
      026A 214 ; RE
      5B FD92 CF DE 026A 215 MOVAL T4FAB,R11 ; r11 will be fab throughout RF
      026F 216 ; initialize values for restartability RI
      026F 217 $FAB_STORE FAB=R11,- RM
      026F 218 SHR=<PUT,GET,UPI>,- RM
      026F 219 FOP=<SUP,CTG>,- RS
      026F 220 XAB=FHCXAB,- RV
      026F 221 ALQ=#0,- RV
      026F 222 FAC=PUT SA
      FE1D CF B4 0289 223 CLRW XABSW_LRL+FHCXAB ST
      028D 224 $XABPRO_STORE XAB=PROXAB,- SY
      028D 225 PRO=<RWED,RWED,RD,RWED>,- SY
      028D 226 UIC=<333,44> SY
      AB AF 0C A0 D0 02A2 228 MOVL XABSL_UIC(R0),UIC SY
      FEFB CF 08 A0 B0 02A7 229 MOVW XABSW_PRO(R0),SAVEPRO ; save it for checking SY
      02AD 230 $XABALL_STORE XAB=ALQXAB,- SY
      02AD 231 ALQ=#10,- SY
      02AD 232 AOP=CTG SY
      02BB 233 $CREATE FAB=R11,- SY
      02BB 234 ERR=REPORT_ERROR SY
      FD33' 30 02CA 235 BSBW ERR SY
      00000000'8F E1 02CD 236 BBC #DEV$V SQD,- T4
      03 40 AB 02D3 237 FABSL_DEV(R11),10$ T4
      03F7 31 02D6 238 BRW MTA T4
      OA FDFD CF D1 02D9 239 10$: CMPL ALQXAB+XABSL_ALQ,#10 ; if mta, skip this TI
      15 18 02DE 240 BGEQ RIGHT TR
      02E0 241 FIELD <ALQ IN XAB ( NOT = DESIRED ALLOC ON CREATE)> UI
      02F5 242 RIGHT: BBC #FAB$V CBT,FABSL FOP(R11),OK1 UI
      15 04 AB 15 E1 02F5 243 FIELD <CBT BIT SET, THEREFORE> VE
      15 04 AB 14 E0 030F 244 BBS #FAB$V CTG,FABSL FOP(R11),OK2 XA
      0314 245 OK1: FIELD <CTG BIT CLEAR, THEREFORE> XA
      0329 246 OK2: BSBW ZERO_XABS XA
      04FB 30 0329 248 $DISPLAY FAB=R11,- XA
      032C 249 ERR=REPORT_ERROR XA
      FCC2' 30 033B 250 BSBW ERR XA
      0501 30 033E 251 BSBW CHECK CR XA
      0341 252 TYPE <OK AFTER CREATE AND DISPLAY> ; check xabs XA
      0370 253 ; XA
      0370 254 ; XA
      0370 255 :extend XA
      0370 256 : XA
      0370 257 : XA
      0370 258 : XA
      FD67 CF 30 D0 0370 259 MOVL #48,XABSL_ALQ+ALQXAB XA
      FD57 CF D4 0375 260 CLRL XABSL_NXT+ALQXAB XA
      24 AB FD4F CF DE 0379 261 MOVAL ALQXAB,FABSL_XAB(R11) XA
      037F 262 $EXTEND FAB=R11,- ; extend file after create XA

```

```

037F 263
038E 264
038E 265
038E 266 ;using alq from xab
038E 267 :
038E 268
30   FD47 FC6F' 30 038E 269      BSBW    ERR
          CF     D1 0391 270      CMPL    ALQXAB+XAB$L_ALQ,#48 ; alq in xab should ret actual alq
          15     18 0396 271      BGEQ    ALQOK
          0398 272      FIELD   <ALQ IN XAB (NOT = DESIRED ALLOC ON EXTEND)>
          03AD 273      ALQOK:
          03AD 274
          03AD 275
          03AD 276 ;undo damage to xab links
          03AD 277 :
          03AD 278
24   AB    FCEF CF    DE 03AD 279      MOVAL   FHGXAB,FAB$L_XAB(R11)
FCEA CF    FD15 CF    DE 03B3 280      MOVAL   ALQXAB,XAB$L_NXT+FHGXAB
FDOF CF    FD2E CF    DE 03BA 281      MOVAL   PROXAB,XAB$L_NXT+ALQXAB
          03C1 282      $CONNECT RAB=T4RAB,-
          03C1 283      ERR=REPORT_ERROR
32   41 8F   FC29' 30 03D4 284      BSBW    ERR
          6E   00 00000000'EF 2C 03D7 285      MOVCS   #0,(SP),#^A/A/,#50,CPYBUF; move 50 a's into cpybuf
          00000000'EF 00000000'EF 03DD 286      MOVAL   CPYBUF,RAB$L_RBF+T4RAB
          00000000'EF 32   B0 03E2 287      MOVW    #50,RAB$W_RSZ+T4RAB
          03F4 288      $PUT    RAB=T4RAB,-
          03F4 289      ERR=REPORT_ERROR
FBF6' 30 0407 290      BSBW    ERR

```

```

040A 292
040A 293 ; take some time to try out flush
040A 294 ;
040A 295 ;
040A 296
040A 297
040A 298
040A 299
0404 30 0420 300
0404 30 0423 301
0404 30 0423 302
0418 30 0434 303
0418 30 0437 304
043A 305
043A 306
043A 307
043A 308
043A 309
FB9A' 30 0463 310
32 00000000'EF B1 0466 311
15 13 046D 312
046F 313
0484 314 RSZ_OK:
00 41 8F 00000000'EF 32 2D 0484 315
6E
15 13 048E 316
048F 317
0491 318 YES:
04A6 319
04A6 320
04A6 321
04A6 322
04A6 323
04A6 324 10$:
FBAA CF D4 04C6 325
04CA 326
04CA 327
FB20' 30 04DD 328
04E0 329
FB0C' 30 04F1 330
04F4 331
0523 332
0523 333
0523 334 ;
0523 335 : all done with flush test
0523 336 :
0523 337

```

\$FLUSH RAB=T4RAB,-
ERR=REPORT_ERROR
BSBW ERR
BSBW ZERO_XABS
\$OPEN FAB=FLUSH_FAB,-
ERR=REPORT_ERROR
BSBW ERR
BSBW CHECK_XABS
\$CONNECT RAB=FLUSH_RAB,-
ERR=REPORT_ERROR
BSBW ERR
\$GET RAB=FLUSH_RAB,-
ERR=REPORT_ERROR
BSBW ERR
CMPW RAB\$W_RSZ+FLUSH_RAB,#50 ; got right rec. size
BEQL RSZ OR
FIELD <RSZ IN RAB>
CMPC5 #50,CPYBUF,/^A/A/,#0,(SP); is record ok?
BEQL YES
FIELD <RECORD>
\$GET FLUSH_RAB ; this should be eof
CMPL R0,#RMSS_EOF
BEQL 10\$
MOVAL FLUSH_RAB,R10
BSBW EOFPUT
CLRL FAB\$L_XAB+FLUSH_FAB ; no xabs on close for now
\$DISCONNECT RAB=FLUSH_RAB,- ; clean up after flush
ERR=REPORT_ERROR
BSBW ERR
\$CLOSE FAB=FLUSH_FAB,- ; all done w/ flush test
ERR=REPORT_ERROR
BSBW ERR
TYPE <ALL DONE WITH FLUSH TEST>

			0523	339		
			0523	340		
			0523	341	\$DISCONNECT RAB=T4RAB,-	
			0523	342	ERR=REPORT_ERROR	
	FAC7'	30	0536	343	BSBW ERR	
			0539	344	\$CLOSE FAB=R11,-	
			0539	345	ERR=REPORT_ERROR	
	FAB5'	30	0548	346	BSBW ERR	
			054B	347		
24 AB	FB4E CF	17 AB	94	348	CLRB FABSB SHR(R11)	
02D0		DE	054E	349	MOVAL FHGXAB,FABSL_XAB(R11) ; set up xab links again	
		30	0554	350	BSBW ZERO XABS	
			0557	351	\$OPEN FAB=R11,-	
	FA97'	30	0566	352	ERR=REPORT_ERROR	
			0569	353	BSBW ERR	
15 04 AB	15	E1	0569	354		
			056E	355	BBC #FAB\$V CBT,FABSL_FOP(R11),CC	
15 04 AB	14	E1	0583	356	FIELD <CBT BIT WAS SET, THEREFORE>	
			0588	357	CC: BBC #FAB\$V CTG,FABSL_FOP(R11),OK; after extend, not ctg	
			0590	358	FIELD <CTG BIT WAS SET, THEREFORE>	
			0590	359		
			0590	360	:	
			0590	361	: check fhc xab	
			0590	362	:	
			0590	363		
	02B2	30	059D	364	OK: BSBW CHECK_XABS	
			05A0	365		
			05A0	366	:	
			05A0	367	: do another extend, forcing it to get the value from the alq of the fab	
			05A0	368	:	
			05A0	369		
10 AB	24 AB	D4	05A0	370	CLRL FAB\$L_XAB(R11)	
OC	OC	D0	05A3	371	MOVL #12,FAB\$L_ALQ(R11)	
			05A7	372	\$EXTEND FAB=R11,-	
	FA47'	30	05B6	373	ERR=REPORT_ERROR	
OC	10 AB	D1	05B9	374	BSBW ERR	
15	18	05BD	375	CMPL FAB\$L_ALQ(R11),#12 ; alq in fab=12		
		05BF	376	BGEQ ALQOKT		
			377	FIELD <ALQ IN FAB (NOT = DESIRED ALLOCATION AFTER EXTEND)>		
			05D4	378	ALQOK1:	
			05D4	379		
			05D4	380	:	
			05D4	381	: change protection and uic on close	
			05D4	382	:	
			05D4	383		
			05D4	384	\$XABPRO_STORE XAB=PROXAB,-	
			05D4	385	PRO=<RWED,RWED,RED,RWED>,-	
			05D4	386	UIC=<222,55>	
FBB5	CF	50	D0	05E9	MOVL R0,FAB\$L_XAB(R11)	: set up xab
FC56	CF	08 A0	B0	05ED	MOVW XAB\$W_PRO(R0),SAVEPRO	: for check
	OC A0	D0	05F3	388	MOVL XAB\$L_UIC(R0),UIC	: ditto
			05F9	389	\$CLOSE FAB=RT1,-	
			05F9	390	ERR=REPORT_ERROR	
24 AB	F9F5'	30	0608	391	BSBW ERR	
	FA91 CF	DE	060B	392	MOVAL FHGXAB,FAB\$L_XAB(R11)	
			0611	393	\$OPEN FAB=R11,-	
			0611	394	ERR=REPORT_ERROR	
			395			: check changes after ext

```

15 04 AB F9DD' 30 0620 396 BSBW ERR
      15 E1 0623 397 BBC #FABSV_CBT,FABSL_FOP(R11),NOCBT
15 04 AB 14 E1 0628 398 FIELD <CBT BIT WAS SET, THEREFORE>
      063D 399 NOCBT: BBC #FABSV_CTG,FABSL_FOP(R11),NOCTG ; shouldn't be ctg, after extend
      0642 400 FIELD <CTG BIT WAS SET, AFTER 2 EXTENDS, THEREFORE>
00000046 59 FA45 CF DE 0657 401 NOCTG: MOVAL FHCXAB,R9 ; check pertinent fields
     8F OC A9 D1 065C 402 CMPL XAB$L_HBK(R9),#70 ; alq=10+48+12
     15 18 0664 403 BGEQ HBKOK
      0666 404 FIELD <HBK IN FHCXAB (AFTER 2ND EXTEND)>
00 28 A9 D1 067B 405 HBKOK: CMPL XAB$L_SBN(R9),#0 ; not ctg anymore
     15 13 067F 406 BEQL STILL_OK
      0681 407 FIELD <SBN IN FHCXAB (AFTER 2ND EXTEND)>
      0696 408 STILL_OK: BSBW CHECK_ALL
01C9 30 0696 410 .BYTE 70,0 ; values for alq,ctg (not ctg)
00 46 0699 411 BSBW CHECK_PRO
025E 30 069B 412 TYPE <DONE WITH 2ND EXTEND, NOW TEST DATES>
24 AB D4 06CD 413 CLRL FABSL_XAB(R11) ; no xabs on this close, for now
      06D0 414 MTA: $CLOSE FAB=RT1,- ; continue if mta
      06D0 415 CLRL FAB=RT1,-
      F91E' 30 06DF 416 BSBW ERR=REPORT_ERROR
      06E2 417 BSBW ERR
      06E2 418 :
      06E2 419 :before finishing up, have some fun with the dat and rdt xabs
      06E2 420 :
      06E2 421 :
      06E2 422
59 FASE CF DE 06E2 423 MOVAL DATXAB,R9
      06E7 424 SBINTIM_S CDTDEC,XAB$Q_CDT(R9)
      06F5 425 SBINTIM_S RDTDEC,XAB$Q_RDT(R9)
      0703 426 SBINTIM_S EDTDEC,XAB$Q_EDT(R9)
08 A9 00C8 8F B0 0711 427 MOVW #200,XABSW_RVN(R9)
FB08 CF 00C8 8F B0 0717 428 MOVW #200,CURRVN
24 AB 59 D0 071E 429 MOVL R9,FABSL_XAB(R11)
      04 A9 D4 0722 430 CLRL XAB$L_NXT(R9)
      FB1F CF FA98 CF DE 0725 431 MOVAL RDT,CURRDAT
      04 AB 00000080 8F C8 072C 432 BISL #FABSM_RWO,FABSL_FOP(R11) ; current rdt str
      0734 433 $CREATE FAB=R1T,- ; rewind if mta
      0734 434 CLRL FAB=RT1,-
      0734 435 BSBW ERR=REPORT_ERROR
      F8BA' 30 0743 435 MOVAL RDTXAB,XAB$L_NXT+DATXAB
      FA26 CF 00C6 30 0746 436 BSBW ZERO_DAT_XABS
      00C6 30 074D 437 $DISPLAY FAB=R11,-
      0750 438 BSBW ERR=REPORT_ERROR
      0750 439 BSBW ERR
      F89E' 30 075F 440 BSBW CHECK_DATES
      032B 30 0762 441 MOVAL RDT2,CURRDAT ; get a new rdt
      59 FA00 CF DE 076C 442 MOVAL RDTXAB,R9
      FA00 CF DE 076C 443 MOVAL RDT2DEC,XAB$Q_RDT(R9)
      0771 444 SBINTIM_S #300,XABSW_RVN(R9)
08 A9 012C 8F B0 077F 445 MOVW #300,CURRVN
FA9A CF 012C 8F B0 0785 446 MOVL R9,FABSL_XAB(R11) ; only rdt for close
24 AB 59 D0 078C 447 $CLOSE FAB=R11,-
      0790 448 MOVAL FAB=RT1,-
      0790 449 BSBW ERR=REPORT_ERROR
      F85E' 30 079F 450 MOVAL DATXAB,FABSL_XAB(R11)
      F99E CF DE 07A2 451 CLRL XAB$L_NXT+DATXAB ; only dat for open
      F99C CF D4 07A8 452

```

RMSTEST4
009

XAB RMS TEST PROGRAM

L 9

16-SEP-1984 01:47:44 VAX/VMS Macro V04-00
5-SEP-1984 04:21:52 [UETP.SRC]RMSTEST4.MAR;1

Page 10
(9)

68	10	07AC	453	BSBB	ZERO_DAT XABS			
		07AE	454	\$FAB_STORE	FAB=R11,-	; can't "put" to mta		
		07AE	455		FAC=GET			
		07B2	456	SOPEN	FAB=R11,-			
		07B2	457		ERR=REPORT_ERROR			
24 AB	F83C'	30	07C1	458	BSBW	ERR		
	F9AB	CF	DE	07C4	459	MOVAL	RDTXAB,FAB\$L_XAB(R11)	; get rdt on display
				07CA	460	\$DISPLAY	FAB=R11,-	
				07CA	461		ERR=REPORT_ERROR	
	F824'	30	07D9	462	BSBW	ERR		
	02B1	30	07DC	463	BSBW	CHECK_DATES		
24 AB	D4	07DF	464	CLRL	FAB\$L_XAB(R11)	; no xabs for this close		
				07E2	465	\$CLOSE	FAB=RT1,-	
				07E2	466		ERR=REPORT_ERROR	
	F80C'	30	07F1	467	BSBW	ERR		
				07F4	468			
				07F4	469			
				07F4	470	\$FAB_STORE	FAB=R11,-	; restore fac
				07F4	471		FAC=PUT	
04 AB	00000080	BF	CA	07F8	472	BICL	#FAB\$M_RWO,FAB\$L_FOP(R11)	; and fop
				0800	473	FINISH	<XAB TESTS>	
				04	0815	RET		

RM
VO

0816 476
0816 477 ;
0816 478 ;2 routines to zero out the xabs before checking the results
0816 479 ;
0816 480
0816 481 ZERO_DAT_XABS:
0816 482
0816 483 ;
0816 484 ;zero out the dat and rdt xabs
0816 485 ;
0816 486
F92C CF 26 00 6E 00 2C 0816 487 MOVCS #0,(SP),#0,#<XAB\$C_DATLEN-6>,DATXAB+6
F950 CF 0E 00 6E 00 2C 081E 488 MOVCS #0,(SP),#0,#<XAB\$C_RDTLEN-6>,RDTXAB+6
0052 8F 00 6E 00 2C 0826 489 RSB
F877 CF 26 00 6E 00 2C 0827 490
F89B CF 1A 00 6E 00 2C 0827 491 ZERO_XABS:
0052 8F 00 6E 00 2C 0827 492
F8B1 CF 05 0837 493 ;
F8B1 CF 05 083E 494 ;zero out the fhc, all and pro xabs
F8B1 CF 05 0841 495 ;
F8B1 CF 05 0841 496
F8B1 CF 05 0841 497 MOVCS #0,(SP),#0,#<XAB\$C_FHCLEN-6>,FHGXAB+6
F8B1 CF 05 0841 498 MOVCS #0,(SP),#0,#<XAB\$C_ALLLEN-6>,ALQXAB+6
F8B1 CF 05 0841 499 MOVCS #0,(SP),#0,#<XAB\$C_PROLEN-6>,PROXAB+6
F8B1 CF 05 0841 500 RSB

```

0842 502 CHECK_CR:
0842 503
0842 504 ;
0842 505 ;routine to check xabs after create and subsequent display
0842 506 ;
0842 507
00F7 30 0842 508     BSBW   CHECK_FHC
01 00 0A 00 0845 509     .BYTE  0,10,0,1      ; values for lrl,alq,ffb,sbn
0016 30 0849 510     BSBW   CHECK_ALL
01 0A 084C 511     .BYTE  10,1      ; values for alq,ctg ( should be)
00AB 30 084E 512     BSBW   CHECK_PRO
05 0851 513     RSB
0852 514
0852 515 CHECK_XABS:
0852 516
0852 517 ;
0852 518 ;general routine to check out all xabs
0852 519 ;
0852 520
00E7 30 0852 521     BSBW   CHECK_FHC
00 38 3A 32 0855 522     .BYTE  50,58,56,0    ; values for lrl,alq,ffb,sbn(not ctg)
0006 30 0859 523     BSBW   CHECK_ALL
00 3A 085C 524     .BYTE  58,0      ; values for alq,ctg ( not ctg anymore)
009B 30 085E 525     BSBW   CHECK_PRO
05 0861 526     RSB
0862 527
0862 528 CHECK_ALL:
0862 529
0862 530 ;
0862 531 ;routine to check out the allocation xab
0862 532 ;
0862 533
59 F866 CF DE 0862 534     MOVAL  ALQXAB,R9
OF 14 A9 B1 0867 535     CMPW   XAB$W_DEQ(R9),#15
15 13 086B 536     BEQL   DEQOK
00 16 A9 91 0882 537     FIELD  <DEQ IN ALL. XAB>
15 13 0886 538     DEQOK: CMPB   XAB$B_BKZ(R9),#0
0888 539     BEQL   BKZOK
50 00 BE 9A 089D 540     FIELD  <BKZ IN ALL. XAB>
6E D6 08A1 541     BKZOK: MOVZBL @(SP),R0
10 A9 50 D1 08A3 542     INCL   (SP)
15 15 08A7 543     CMPL   R0,XAB$L_ALQ(R9)
08A9 544     BLEQ   ALQOK2
50 00 BE 9A 08BE 545     FIELD  <ALQ IN ALL. XAB>
6E D6 08C2 546     ALQOK2: MOVZBL @(SP),R0
1A 50 E9 08C4 547     INCL   (SP)
2F 08 A9 07 E0 08C7 548     BLBC   R0,NOTCTG
08CC 549     BBS    #XAB$V CTG,XAB$B_AOP(R9),AOPOK ; should be set
15 08 A9 07 E1 08E1 550     FIELD  <CTG C|R IN AOP, THEREFORE>
08E6 551     NOTCTG: BBC   #XAB$V CTG,XAB$B_AOP(R9),AOPOK ; should be clear
05 08FB 552     FIELD  <CTG SET IN AOP, THEREFORE>
08FC 553     AOPOK: RSB
08FC 554
08FC 555     CHECK_PRO:
08FC 556
08FC 557 ;
08FC 558 ;check the protection xab

```

RMSTEST4
009

XAB RMS TEST PROGRAM

B 10

16-SEP-1984 01:47:44 VAX/VMS Macro V04-00
5-SEP-1984 04:21:52 [UETP.SRC]RMSTEST4.MAR;1

Page 13
(12)

			08FC	559 :		
			08FC	560		
08	A9	F7EC CF	DE	08FC	561	MOVAL PROXAB,R9
		F8A3 CF	15	B1	0901	CMPW SAVEPRO,XAB\$W_PRO(R9) ; cmp to saved value
			13	0907	562	BEQL PROOK
0C	A9	F92D CF	D1	091E	563	FIELD <PROT FIELD IN PROT XAB>
		15	13	0924	564	CMPL UIC,XAB\$L_UIC(R9)
				0926	565	BEQL UICOK
				05	093B	FIELD <UIC FIELD IN PROT. XAB>
				093C	568	UICOK: RSB
					569	

RM
VC

```

093C 571 CHECK_FHC:
093C 572
093C 573 ; check fhc xab carefully
093C 574
093C 575
093C 576

59 F760 CF DE 093C 577 MOVAL FHCXAB,R9 ; r9 is ptr to xab thru-out cmp's
0941 578 EXTC: CMPB XAB$B_RFO(R9),#FABSC_VFC; check rec. format & org.
03 08 A9 91 0941 579 BEQL RFOC
15 13 0945 580 FIELD <RFO IN FHC XAB>
0947 581 RF0C: BITB #FAB$M_CR,XAB$B_ATR(R9) ; check rat field
15 12 0960 582 ATRC: BNEQ ATRC
09 A9 02 93 095C 583 FIELD <ATR IN FHC XAB>
15 12 0962 584 MOVZBL @(SP),R0
0977 585 ATRC: INCL (SP)
6E D6 097B 586 CMPW XAB$W_LRL(R9),R0 ; check longest record len
0A A9 B1 097D 587 BEQL LRLC
15 13 0981 588 FIELD <LRL IN FHC XAB>
0983 589 MOVZBL @(SP),R0
0998 590 LRLC: INCL (SP)
6E D6 099C 591 CMPL XAB$L_HBK(R9),R0 ; check alq
15 18 09A2 592 BGEQ HBKC
09A4 593 FIELD <HBKC IN FHC XAB>
10 A9 D1 09B9 594 CMPL XAB$L_EBK(R9),#1 ; check end block
15 13 09BD 595 BEQL EBKC
09BF 596 FIELD <EBKC IN FHC XAB>
09D4 597 MOVZBL @(SP),R0
6E D6 09D8 598 INCL (SP)
14 A9 B1 09DA 600 CMPW XAB$W_FFB(R9),R0 ; check first free byte
15 13 09DE 601 BEQL FFBC ; its len of rec + fsz + 2
09E0 602 FIELD <FFB IN FHC XAB>
00 16 A9 91 09F5 603 FFBC: CMPB XAB$B_BKZ(R9),#0 ; check bucket size
15 13 09F9 604 BEQL BKZC
09FB 605 FIELD <BKZ IN FHC XAB>
04 17 A9 91 0A10 606 BKZC: CMPB XAB$B_HSZ(R9),#4 ; check fixed area size
15 13 0A14 607 BEQL HSZC
0A16 608 FIELD <HSZ IN FHC XAB>
0064 8F 18 A9 B1 0A2B 609 HSZC: CMPW XAB$W_MRZ(R9),#100 ; check max. rec size
15 13 0A31 610 BEQL MRZC
0A33 611 FIELD <MRZ IN FHC XAB>
0F 1A A9 B1 0A48 612 MRZC: CMPW XAB$W_DXQ(R9),#15 ; check def ext. qty
15 13 0A4C 613 BEQL DXQC
0A4E 614 FIELD <DXQ IN FHC XAB>
50 00 BE 9A 0A63 615 DXQC: MOVZBL @(SP),R0
6E D6 0A67 616 INCL (SP)
08 50 E9 0A69 617 BLBC R0,10$ ; make sure non-zero lbn
00 28 A9 D1 0A6C 618 CMPL XAB$L_SBN(R9),#0
0A70 619
0A70 620 ; since it's ctg
0A70 621 :
0A70 622 :
0A70 623
1D 12 0A70 624 BNEQ FHC_OK
06 11 0A72 625 BRB 20$
00 28 A9 D1 0A74 626 10$: CMPL XAB$L_SBN(R9),#0 ; make sure zero lbn
15 13 0A78 627 BEQL FHC_OR ; since it isn't contig.

```

RMSTEST4
009

XAB RMS TEST PROGRAM

D 10

16-SEP-1984 01:47:44 VAX/VMS Macro V04-00
5-SEP-1984 04:21:52 [UETP.SRC]RMSTEST4.MAR;1

Page 15
(13)

05 0A7A 628 20\$: FIELD <SBN IN FHC XAB>
05 0A8F 629 FHC_OK: RSB

```

0A90 631
0A90 632 CHECK_DATES:
0A90 633
0A90 634 :
0A90 635 ;routine to check edt and cdt in dat xab,
0A90 636 ;and rdt and rvn in both dat and rdt xab's
0A90 637 :
0A90 638

05 40 AB F793 CF 0B B0 0A90 639      MOVW #DATLEN,LEN          : default is check date only
00000000'8F E0 0A95 640      BBS #DEV$V_SQD,FABSL_DEV(R11),10$ ; if not mta, check date and time
F785 CF 14 B0 0A9E 641      MOVW #TIMLEN,LEN
59 F69D CF DE 0AA3 642 10$: MOVAL DATXAB,R9
F766 CF F6E9 CF F76A CF 29 0ABA 643      $ASCTIM_S ,CMPDATDEC,XAB$Q_CDT(R9)
15 13 0AC4 644      CMPC3 LEN,CDT,CMPDAT
0AC6 645      BEQL CDTOK
OADB 646      FIELD <CDT IN DAT XAB>
OADB 647 CDTOK:
OADB 648      $ASCTIM_S ,CMPDATDEC,XAB$Q_EDT(R9)
F735 CF F6E6 CF 0B 29 0AED 649      CMPC3 #DATLEN,EDT,CMPDAT ; only check date
15 13 0AF5 650      BEQL EDTOK
0AF7 651      FIELD <EDT IN DAT XAB>
01 40 AB 00000000'8F E1 0B0C 652 EDTOK:
05 0B15 653      BBC #DEV$V_SQD,FABSL_DEV(R11),10$ ; that's it if mta
F6F8 CF F71C DF F6FC CF 29 0B28 654 10$: $ASCTIM_S ,CMPDATDEC,XAB$Q_RDT(R9)
15 13 0B32 655      CMPC3 LEN,@CURRDT,CMPDAT
0B34 656      BEQL RDTOK
0B49 657      FIELD <RDT IN DAT XAB>
0B49 659 RDTOK:
0B5C 660      $ASCTIM_S ,CMPDATDEC,XAB$Q_RDT+RDTXAB
F6C4 CF F6E8 DF F6C8 CF 29 0B66 661      CMPC3 LEN,@CURRDT,CMPDAT
15 13 0B68 662      BEQL RDTOK1
0B7D 663      FIELD <RDT IN RDT XAB>
08 A9 F6A5 CF B1 0B7D 664 RDTOK1:
15 13 0B83 665      CMPW CURRVN,XAB$W_RVN(R9)
0B85 666      BEQL RVNOK
F5D7 CF F688 CF B1 0B9A 667 RVNOK: FIELD <RVN IN DAT XAB>
15 13 0BA1 668      CMPW CURRVN,XAB$W_RVN+RDTXAB
0BA3 669      BEQL RVNOK1
0BB8 670      FIELD <RVN IN RDT XAB>
05 0BB9 671 RVNOK1: RSB
0BB9 672 .END

```

\$\$._PSECT_EP	= 00000000		ERR	***** X 01
\$\$._TAB	= 00000184	R D 01	EXTC	00000941 R D 01
\$\$._TABEND	= 000001A8	R D 01	EXTRA	= 00000024
\$\$._TMP	= 00000001		FAB\$B_FAC	= 00000016
\$\$._TMP1	= 00000002		FAB\$B_FNS	= 00000034
\$\$._TMP2	= 0000005B		FAB\$B_SHR	= 00000017
\$\$._TMP5	= 00000002		FAB\$C_BID	= 00000003
\$\$._TMPX	= 00000352	R D 04	FAB\$C_BLN	= 00000050
\$\$._TMPX1	= 0000000E		FAB\$C_SEQ	= 00000000
\$\$._RMSTEST	= 0000001E		FAB\$C_VAR	= 00000002
\$\$._RMS_PBUGCHK	= 00000010		FAB\$C_VFC	= 00000003
\$\$._RMS_TBUGCHK	= 00000008		FAB\$L_ALQ	= 00000010
\$\$._RMS_UMODE	= 00000004		FAB\$L_DEV	= 00000040
..AFLG	= 00000000	D	FAB\$L_FNA	= 00000020
..FLG	= 00000002	D	FAB\$L_FOP	= 00000004
..MOD	= 00000001	D	FAB\$L_XAB	= 00000024
..N	= 00000001		FAB\$M_CR	= 00000002
..TYP	= 00000003	D	FAB\$M_RWO	= 00000080
.LEN	= 00000001	D	FAB\$V_CBT	= 00000015
ALQOK	000003AD	R D 01	FAB\$V_CHAN_MODE	= 00000002
ALQOK1	000005D4	R D 01	FAB\$V_CR	= 00000001
ALQOK2	000008BE	R D 01	FAB\$V_CTG	= 00000014
ALQXAB	000000CC	RG D 01	FAB\$V_FILE_MODE	= 00000004
AOPOK	000008FB	R D 01	FAB\$V_GET	= 00000001
ATRC	00000977	R D 01	FAB\$V_LNM_MODE	= 00000000
BEGPUT	***** X 01		FAB\$V_PUT	= 00000000
BEG_DESCR	***** X 01		FAB\$V_SUP	= 00000002
BKZC	00000A10	R D 01	FAB\$VUPI	= 00000006
BKZOK	0000089D	R D 01	FAB\$W_GBC	= 00000048
CC	00000583	R D 01	FFBC	000009F5 R D 01
CDT	000001AA	R D 01	FHC_XAB	000000A0 RG D 01
CDTDEC	00000206	R D 01	FHC_OK	00000A8F R D 01
CDTL	= 00000017	D	FINPUT	***** X 01
CDTOK	00000ADB	R D 01	FIN_DESCR	***** X 01
CHECK_ALL	00000862	R D 01	FLDPUT	***** X 01
CHECK_CR	00000842	R D 01	FLD_DESCR	***** X 01
CHECK_DATES	00000A90	R D 01	FLUSH_FAB	00000050 RG D 01
CHECK_FHC	0000093C	R D 01	FLUSH_RAB	***** X 01
CHECK_PRO	000008FC	R D 01	HBKC	000009B9 R D 01
CHECK_XABS	00000852	R D 01	HBKOK	0000067B R D 01
CMDORAB	***** X 01		HSZC	00000A2B R D 01
CMPDAT	0000022A	R D 01	LEN	00000228 R D 01
CMPDATDEC	00000243	R D 01	LRLC	00000998 R D 01
CPYBUF	***** X 01		MRZC	00000A48 R D 01
CURRDT	0000024B	R D 01	MTA	000006D0 R D 01
CURRVN	00000226	R D 01	NAMBLK	***** X 01
DATLEN	= 0000000B		NOCBT	0000063D R D 01
DATXAB	00000144	RG D 01	NOCTG	00000657 R D 01
DEQOK	00000882	R D 01	NOTCTG	000008E1 R D 01
DEV\$V_SQD	***** X 01		OK	0000059D R D 01
DXQC	00000A63	R D 01	OK1	0000030F R D 01
EBKC	000009D4	R D 01	OK2	00000329 R D 01
EDT	000001D8	R D 01	PROOK	0000091E R D 01
EDTDEC	00000216	R D 01	PROXAB	000000EC RG D 01
EDTL	= 00000017		RAB\$L_RBF	***** X 01
EDTOK	00000B0C	R D 01	RAB\$W_RSZ	***** X 01
EOFPUT	***** X 01		RDT	000001C1 R D 01

RMS TEST4 Symbol table

XAB RMS TEST PROGRAM

G 10

16-SEP-1984 01:47:44 VAX/VMS Macro V04-00
5-SEP-1984 04:21:52 [CUETP.SRC]RMSTEST4.MAR;1

Page 18
(15)

RDT2	000001EF	R	D	01
RDT2DEC	0000021E	R	D	01
RDTDEC	0000020E	R	D	01
RDTL	= 00000017	D	D	
RDTL2	= 00000017	D	D	
RDTOK	00000B49	R	D	01
RDTOK1	00000B7D	R	D	01
RDTXAB	00000170	RG	D	01
REPORT_ERROR	*****	X		01
RFOC	0000095C	R	D	01
RIGHT	000002F5	R	D	01
RMSS_EOF	*****	X		01
RMTSTEST_4A	00000253	RG	D	01
RSZ_OK	00000484	R	D	01
RVNOK	00000B9A	R	D	01
RVNOK1	00000BB8	R	D	01
SAVEPRO	000001A8	R	D	01
STILL_OK	00000696	R	D	01
SYSS\$ACTIM	*****	GX		01
SYSS\$BINTIM	*****	GX		01
SYSS\$CLOSE	*****	GX		01
SYSS\$CONNECT	*****	GX		01
SYSS\$CREATE	*****	GX		01
SYSS\$DISCONNECT	*****	GX		01
SYSS\$DISPLAY	*****	GX		01
SYSS\$EXTEND	*****	GX		01
SYSS\$FLUSH	*****	GX		01
SYSS\$GET	*****	GX		01
SYSS\$OPEN	*****	GX		01
SYSS\$PUT	*****	GX		01
T4FAB	00000000	RG	D	01
T4RAB	*****	X		01
T4START	00000000	RG	D	01
TIMLEN	= 00000014	D	D	
TRMXAB	00000184	RG	D	01
UIC	0000024F	R	D	01
UICOK	00000938	R	D	01
VERBOSITY	*****	X		01
XAB\$B_AID	= 00000017	D		
XAB\$B_AOP	= 00000008	D		
XAB\$B_ATR	= 00000009	D		
XAB\$B_BKZ	= 00000016	D		
XAB\$B_HSZ	= 00000017	D		
XAB\$B_MTACC	= 0000000A	D		
XAB\$B_PROT_MODE	= 00000010	D		
XAB\$B_PROT_OPT	= 0000000B	D		
XAB\$B_RFO	= 00000008	D		
XAB\$C_ALL	= 00000014	D		
XAB\$C_ALLLEN	= 00000020	D		
XAB\$C_DAT	= 00000012	D		
XAB\$C_DATLEN	= 0000002C	D		
XAB\$C_FHC	= 0000001D	D		
XAB\$C_FHCLEN	= 0000002C	D		
XAB\$C_PRO	= 00000013	D		
XAB\$C_PROLEN	= 00000058	D		
XAB\$C_RDT	= 0000001E	D		
XAB\$C_RDTLEN	= 00000014	D		

```
! Psect synopsis !
```

PSECT name	Allocation	PSECT No.	Attributes	CON	ABS	LCL	NOSHR	NOEXE	NORD	NOWRT	NOVEC	BYTE
. ABS	00000000	(0.)	00 (0.)	NOPIC	USR	CON	ABS	GBL	NOSHR	EXE	RD	WRT NOVEC LONG
RMTEST	00000BB9	(3001.)	01 (1.)	NOPIC	USR	CON	REL	LCL	NOSHR	EXE	RD	WRT NOVEC BYTE
\$ABSS	00000000	(0.)	02 (2.)	NOPIC	USR	CON	ABS	LCL	NOSHR	EXE	RD	WRT NOVEC BYTE
\$RMSNAM	0000002A	(42.)	03 (3.)	NOPIC	USR	CON	REL	LCL	NOSHR	EXE	RD	WRT NOVEC BYTE
--\$RMSNAM	00000360	(864.)	04 (4.)	NOPIC	USR	CON	REL	LCL	NOSHR	EXE	RD	WRT NOVEC BYTE

```
! Performance indicators !
```

Phase	Page faults	CPU Time	Elapsed Time
Initialization	29	00:00:00.09	00:00:00.43
Command processing	105	00:00:00.63	00:00:02.75
Pass 1	362	00:00:17.15	00:00:38.17
Symbol table sort	0	00:00:00.53	00:00:01.16
Pass 2	148	00:00:03.68	00:00:08.25
Symbol table output	24	00:00:00.15	00:00:00.31
Psect synopsis output	3	00:00:00.02	00:00:00.30
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	673	00:00:22.25	00:00:51.37

The working set limit was 1350 pages.

80832 bytes (158 pages) of virtual memory were used to buffer the intermediate code.

There were 30 pages of symbol table space allocated to hold 460 non-local and 9 local symbols.

672 source lines were read in Pass 1, producing 53 object records in Pass 2.

67 pages of virtual memory were used to define 50 macros.

```
! Macro library statistics !
```

Macro library name	Macros defined
\$255\$DUA28:[SYS.OBJ]LIB.MLB:1	0
\$255\$DUA28:[SYSLIB]STARLET.MLB:2	50
TOTALS (all libraries)	50

1077 GETS were required to define 50 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LISS:RMSTEST4/OBJ=OBJ\$:RMSTEST4 MSRC\$:RMSTEST4/UPDATE=(ENH\$:RMSTEST4)+EXECMLS/LIB

0409 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

